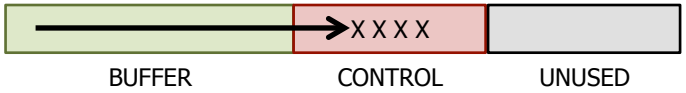
Space/Time Analysis for Cybersecurity (STAC)
Proposer's Day
DARPA-BAA-14-60

Tim Fraser
Program Manager
Information Innovation Office (I2O)
DARPA

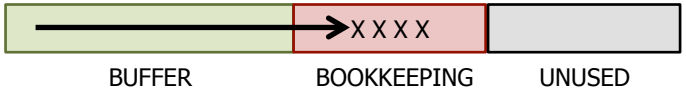
22 September, 2014

Past: Flawed Implementations of Algorithms

STACK CORRUPTION

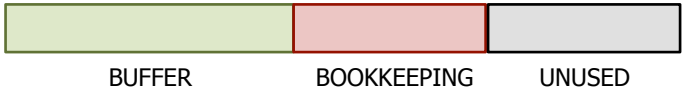


HEAP CORRUPTION

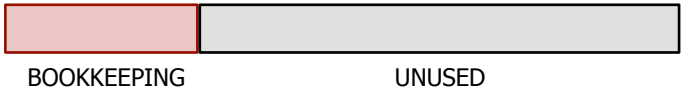


USE AFTER FREE

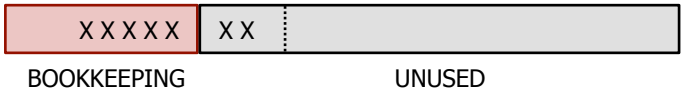
1.

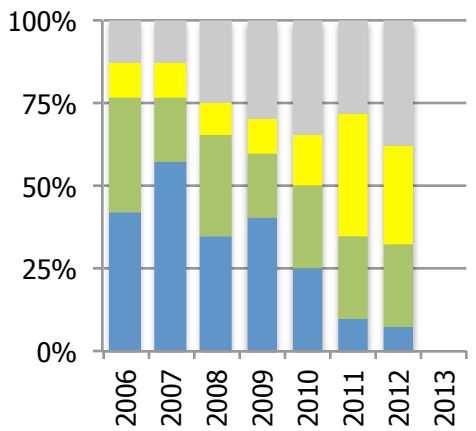


2.



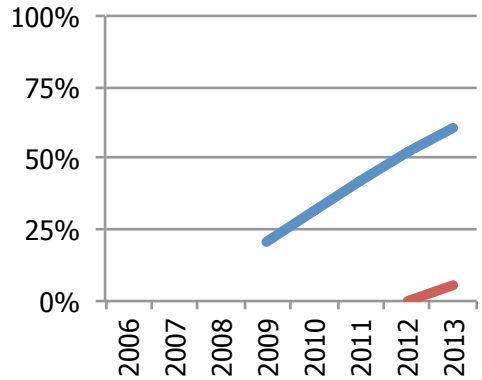
3.





Prevalence of vulnerability classes exploited

- Others
- Use After Free
- Heap Corruption
- Stack Corruption



Prevalence of defenses in MS Windows

- Address Space Layout Randomization + Data Execution Prevention
- Safe unlinking + Reference count hardening

SOURCES: Microsoft 2013 Software Vulnerability Exploitation Trends and NETMARKETSHARE.COM

Exploitation trends evolve in response to defenses.
Commodity systems now deploy mitigations for common implementation flaws.

Future: Flaws in the Algorithms Themselves

Program Focus: Algorithmic resource usage vulnerabilities.

ALGORITHMIC COMPLEXITY ATTACKS

Small worst-case input causes a crippling space or time usage.

2011

Huge portions of the Web vulnerable to hashing denial-of-service attack

A flaw common to most popular Web programming languages can be used to launch …

by **Jon Brodkin** - Dec 28 2011, 2:25pm EST — ARS TECHNICA

SIDE CHANNEL ATTACKS

Adversary deduces secrets by observing minute differences in space or time used.

2013

Gone in 30 seconds: New attack plucks secrets from HTTPS-protected pages

Exploit called BREACH bypasses the SSL crypto scheme protecting millions of sites.

by **Dan Goodin** - Aug 1 2013, 11:30am EDT — ARS TECHNICA

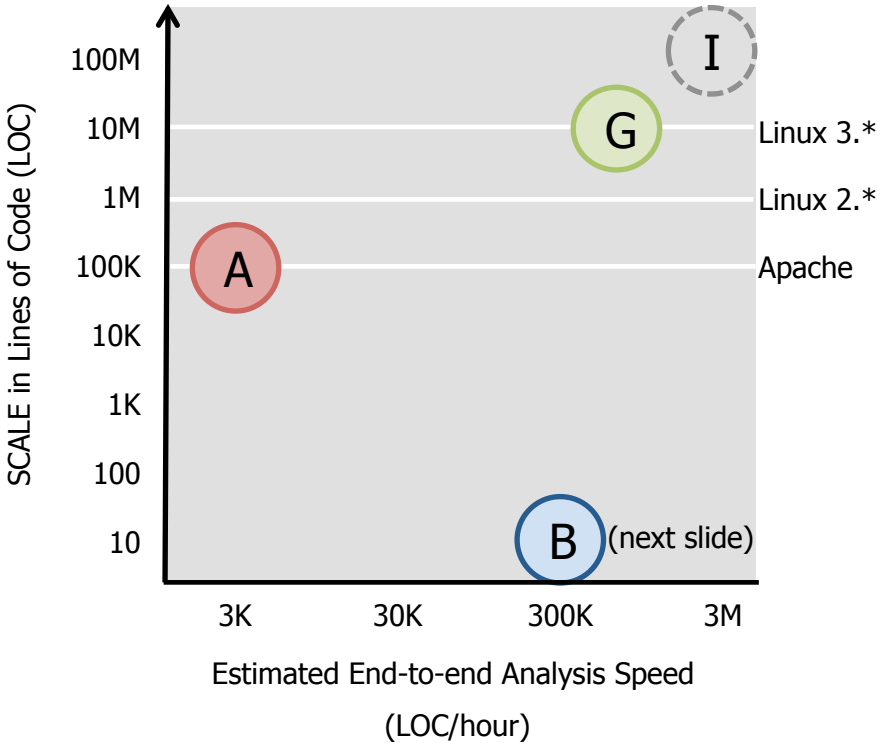
Resource usage vulnerabilities have been reported in:

Search Engines [CHE 10]	Web Browser [PAU 12]	HTTP Secure (HTTPS) [PRA 13]	Programming Languages [WAL 11]	Online health, tax, and investment apps [CHE 10]

Future algorithmic flaws do not involve traditional implementation flaws, are not mitigated by traditional defenses, and thus require a different analysis.

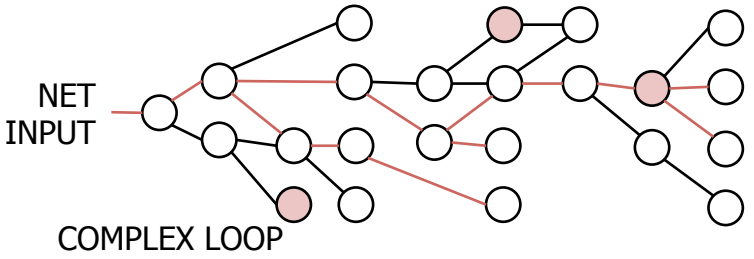
Progress Can Be Made on Scale

Analysis Scale vs. End-to-end Analysis Time



A

Chang and others (UT Austin) 2009



1. Determine which loops are controlled by network input.

   Method: data-flow, control-flow analyses
2. Rank warnings by complexity.

   Method: structural heuristics

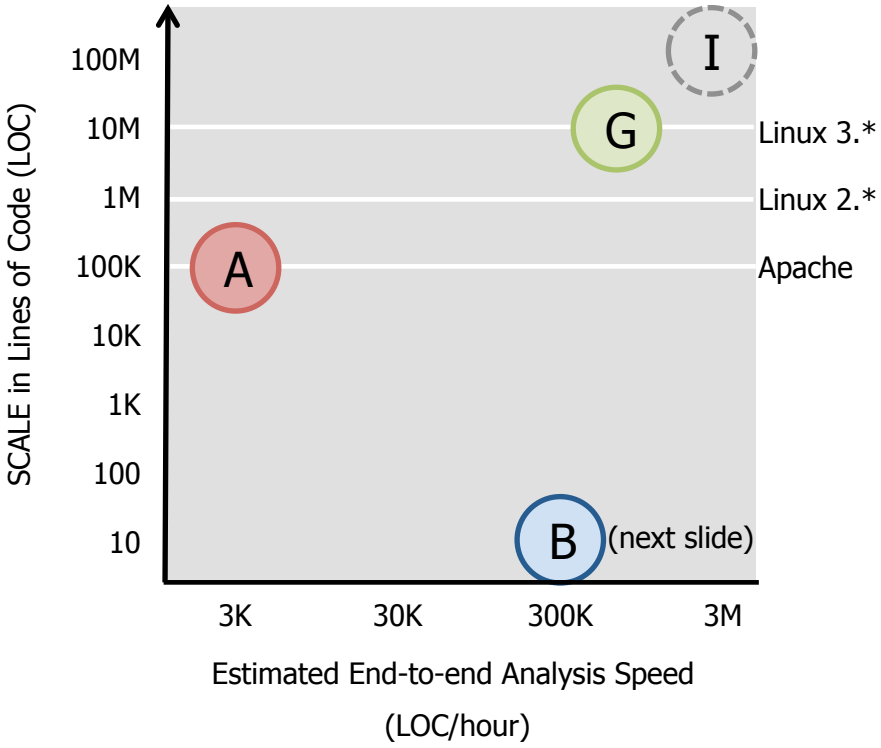
Found vulnerabilities in:

- Expat XML parser (12KLOC)
- WU-FTPD (20KLOC)
- SQLite database engine (63KLOC)

82% false alarm rate.

Progress Can Be Made on Speed

Analysis Scale vs. End-to-end Analysis Time



B

Gulwani & Zuleger (MSR, TU Vienna) 2010

```
Ex1(uint n, bool[] A)
i := 0;
while (i < n)
    j := i + 1;
    while (j < n)
        if (A[j])
            ConsumeResource();
            j--;
            n--;
        j++;
    i++;
```

How many times is this point visited?

1. Extract logic that controls loops

   Method: abstract interpretation

   Max(0, n – j, n – i – 2) ∧ i ≥ 0 ∧ j ≥ 1

2. Compute bounds in terms of input

   Method: constraint solving

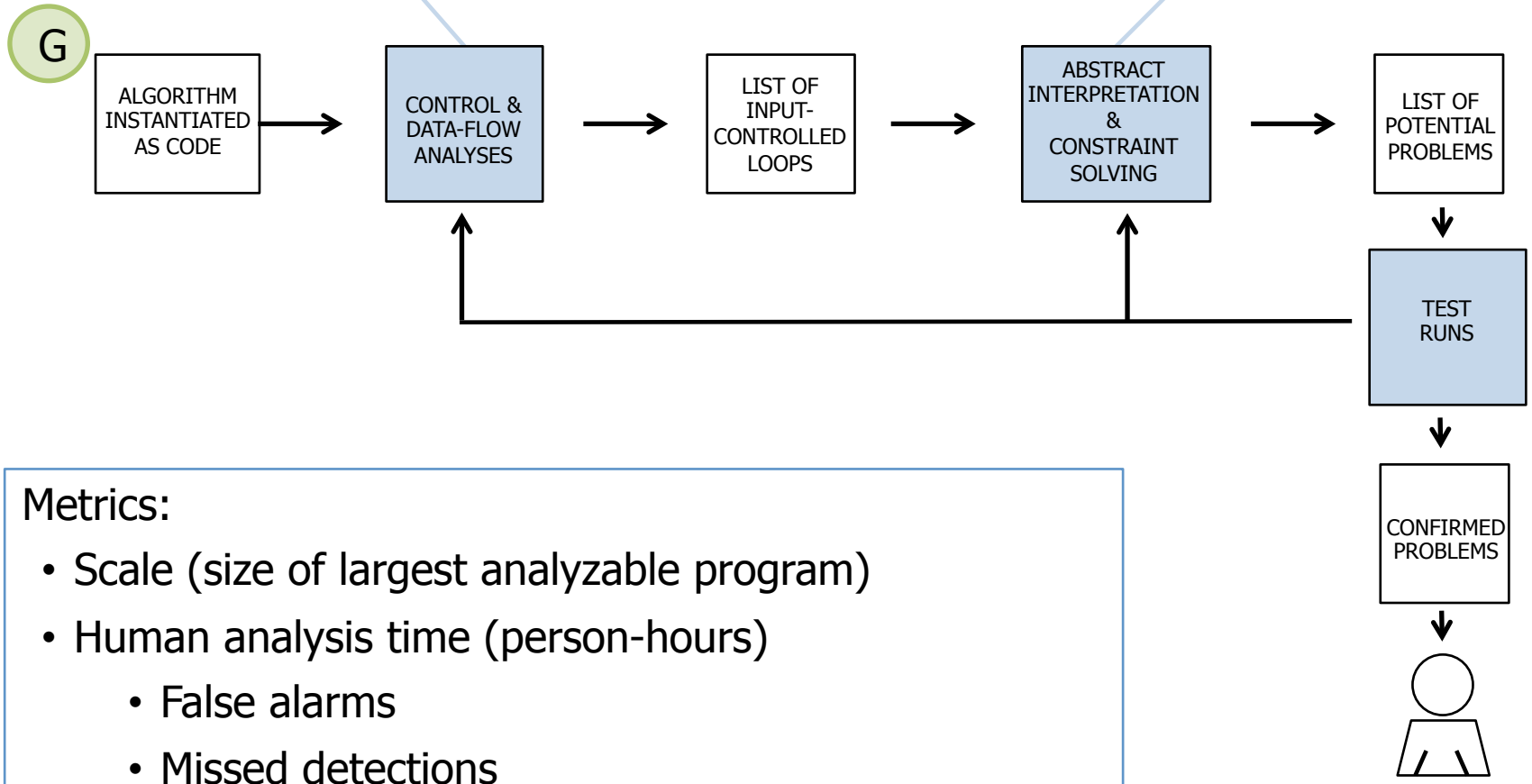
   At most n visits to ConsumeResource()

Computed bounds for complex loops in .Net base-class libraries.

One Plausible Solution Strategy

Research question #1:
What paths exist between inputs and variables, secrets and outputs?

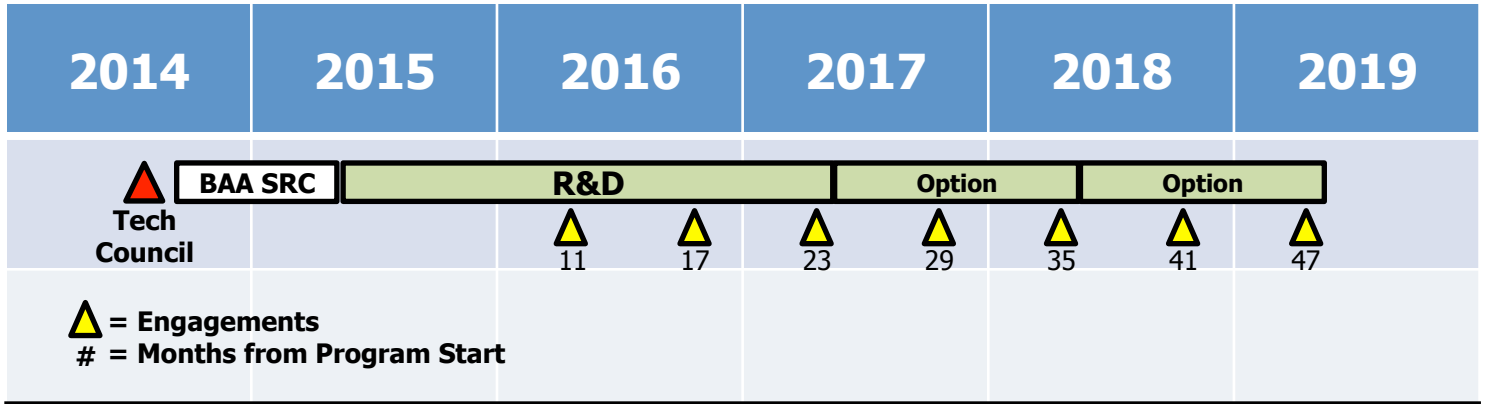
Research question #2:
How do inputs impact resource usage?



Metrics:

- Scale (size of largest analyzable program)
- Human analysis time (person-hours)
  - False alarms
  - Missed detections

Program Schedule and Structure



TA1 – Program Analysis Research & Development (R&D) Teams.

TA2 – Adversarial Challenge Teams.

TA3 – Experimentation Lead: Measure progress with engagements that challenge R&D teams to find space-time vulnerabilities planted in software.

Target software: Java bytecode. No source.

www.darpa.mil

References

[CHE10] Shuo Chen and others. "Side-Channel Leaks in Web Applications: a Reality Today, a Challenge Tomorrow," IEEE Symposium on Security and Privacy, 2010.

[DUO12] Thai Duong and Julianno Rizzo. "The CRIME Attack," Ekoparty, 2012.

[PAU12] Paul. "Leaking information with timing attacks on hashtables, part 1," http://gdtr.wordpress.com/2012/08/07/leaking-information-with-timing-attacks-on-hashtables-part-1/, 2012.

[PRA13] Angelo Prado, Neal Harris, and Yoel Gluck. "SSL, Gone in 60 Seconds – A BREACH beyond CRIME," Black Hat, 2013.

[WAL11] Julian Walde and Alexander Klink, "Effective Denial of Service attacks against web application platforms," Chaos Communications Congress 28C3, 2011.